**СПЕЦИФІКАЦІЯ ПРОГРАМНОГО ПРОДУКТУ**

**Назва продукту:** Тестовий застосунок для бронювання місць з використанням різних підходів до управління станом у React

**Автор:** Малікова Тетяна Віталіївна, група ІПЗздм-23-1

**Керівник:** Смеляков Кирило Сергійович – професор, доцент кафедри програмної інженерії

**Рік:** 2025

**1. Призначення**

Програмний продукт створено з метою демонстрації та порівняння ефективності чотирьох методів управління станом у React-застосунках:

* Redux;
* Zustand;
* Context API;
* Recoil.

Кожна версія реалізує однакову функціональність: бронювання вільних місць, відображення списку заброньованих, скасування бронювання.

Продукт використовується як інструмент експериментального дослідження в дипломній роботі магістра.

**2. Основні функції**

* Візуалізація сітки місць (наприклад, 5×10);
* Вибір/зняття бронювання місць;
* Вивід списку заброньованих місць;
* Кнопка "Забронювати";
* Кнопка "Скасувати";
* Реактивне оновлення інтерфейсу згідно з поточним станом;
* Реалізація 4 окремих версій SPA з різними бібліотеками для управління станом.

**3. Вимоги до середовища**

* **Платформа:** Node.js ≥ 18.x, браузер із підтримкою ES6+
* **Середовище розробки:** Visual Studio Code
* **Менеджер пакетів:** npm
* **Бібліотеки:**
  1. react, react-dom;
  2. redux, @reduxjs/toolkit, react-redux;
  3. zustand;
  4. recoil;
  5. tailwindcss (для стилізації);
  6. react-icons (необов’язково).

**4. Архітектура**

Проєкт реалізовано у вигляді компонентного застосунку React. Кожна версія реалізована у власній директорії. Усі версії мають подібну структуру компонентів:

* App.js — головний компонент;
* SeatGrid.js, Seat.js — для візуалізації місць;
* BookedList.js, BookButton.js — для управління діями.

Стан керується відповідно до обраної бібліотеки — через Redux store, Zustand store, React Context або Recoil atoms/selectors.

**5. Інструкція з використання**

1. Клонувати відповідний репозиторій з GitHub;
2. У директорії відповідної версії виконати команду:
   1. npm install;
   2. npm start;
3. У браузері відкрити http://localhost:3000;
4. Виконати вибір місць, забронювати, перевірити список;
5. Для переходу між версіями використати окремі гілки репозиторію або окремі каталоги.

**6. Вимоги до якості**

* Реактивне оновлення UI без перезавантаження;
* Мінімальна кількість повторних рендерів (оптимізація);
* Продуктивність перевірена інструментами DevTools (Profiler, Memory, Performance);
* Відсутність помилок під час інтерактивної взаємодії;
* Простий, зрозумілий інтерфейс.

**7. Ліцензія та авторські права**

Програмний продукт створено як частина дипломного проєкту. Поширюється як відкритий код (Open Source) з освітньою метою. Авторські права належать автору.